Green IT and Green Software

Verdecchia, Roberto; Lago, Patricia; Ebert, Christof; De Vries, Carol

published in
IEEE Software
2021

DOI (link to publisher)
10.1109/MS.2021.3102254

document version
Peer reviewed version

Accepted version.
Cite as:
Verdecchia, R., Lago, P., Ebert, C., & De Vries, C. (2021). Green IT and Green Software. IEEE Software, 38(6), 7-15. https://doi.org/10.1109/MS.2021.3102254

Green IT and Green Software

Roberto Verdecchia, Vrije Universiteit Amsterdam
Patricia Lago, Vrije Universiteit Amsterdam, and Chalmers University of Technology
Christof Ebert, Vector
Carol de Vries, PhotonDelta

Ecologic behavior is the need across the world to mitigate impacts of climate change. Software and IT play a pivotal role towards ecologic behaviors for many reasons. Being aware that IT systems alone already consume 10% of global electricity, the leading software practitioners must embark on green IT and green coding. Read in this article about hands-on guidance how you can contribute towards a more ecologic software. I look forward to hearing from you about this column and the technologies that matter most for your work.

—Christof Ebert

DOI Keywords: Green IT, Green Coding, Ecology, Cloud Services, Software Technology, Tools

Software and IT usage is continuously growing to keep our society active and manage our individual life. But as they grow, their energy demand is exploding. By 2030, data centers alone will already consume some 10% of global electricity consumption [5]. Including the Internet, telecommunication, and embedded devices the energy consumption will be one third of global demand. Understanding that end-users only consume what we offer, it is the community of software developers who must get active towards ecologic behaviors. Green IT is the call of today. Each single line of code which we develop today may run years from now still on zillions of processors, eating energy and contributing to global climate change.

Green IT and green coding are describing a paradigm switch how software engineers, developers, testers, and IT administrators can make their solutions and services more energy efficient. It is each single software person who contributes. We will give in this article hands-on guidance how to reduce the energy waste of your software, and thus contribute to more ecologic behaviors.

Green IT

With the introduction of high bandwidth data transfers, affordable data plans, the generalized migration to the cloud of software applications and data management, the wide usage of streaming services, and not the least the many embedded computers in our every-day life, digital infrastructures are experiencing an ever-growing demand of energy. While digital transformation looks impressive from an economic perspective, it has its downsides on the ecologic footprint of these businesses.

An immediate action is to adopt more renewable energy. Energy-hungry companies such as Microsoft, Google, and Amazon are currently investing in water energy such as for cooling their data centers, solar energy, and wind farms. Many companies engage in trading CO2 certificates to give a green color to the energy waste of their data centers. But renewable energy only means curing the symptoms. It does not really address reducing the need for energy. To address root causes, we need green IT, such as lower energy consumption in data centers and for cloud services, and green code, which addresses how to efficiently organize the software itself.

Green IT provides solutions to sustainably reduce the energy need of data centers and cloud services. As an example, the Lower Energy Acceleration Program (LEAP) has been launched to explore alternative solutions towards a sustainable growth of the data center industry. As one of our goals, we want to develop a technology landscape for energy efficient digital infrastructures.

In a first step, we conducted a series of interviews and focus groups with various digital infrastructure stakeholders, like data centers, cloud service providers, and cloud customers from both public- and private sector. The participants were asked (i) to indicate and describe the solutions that in their experience will contribute to the sustainability of digital infrastructures, and (ii) to map each solution on different temporal horizons according to the related perceived level of readiness in terms of widespread-adoption and full impact. The resulting temporal horizons are:

- Horizon 1 (H1): State of the art, i.e., solutions for today
- Horizon 2 (H2): Within the next 4-6 years, i.e., solutions for near future
- Horizon 3 (H3): Beyond 6 years, i.e., further evolution

We also identified four types of solutions, namely technical, social, and environmental solutions, and paradigm shifts. An overview of the solution landscape is presented in Figure 1 and described in further detail in [1].

Short-term solutions are readily available for adoption. In Fig. 1 we group the as short-term horizon H1. It includes namely moving to the cloud (i.e., relocating data, computational, and software capabilities from on premise to the cloud). With the popularization of cloud native and serverless applications, resources are accessed on demand, profiting from hyperscale datacenter energy efficiency optimizations. Such solutions entail the use of green energy sources (e.g., solar farms), heuristics for hyperscale hardware management (e.g., reuse of dissipated heat), deployment of domain-specific hardware (e.g., AI accelerators), and tight collaborations between software and hardware companies to develop dedicated hardware components (integrated infrastructures). H1 also sees the rise of energy-aware software optimizations, such as energy efficient use-on-demand, and energy-aware management of idle servers. From a social perspective, communication raises awareness on the environmental impact of personal digital infrastructure usage.

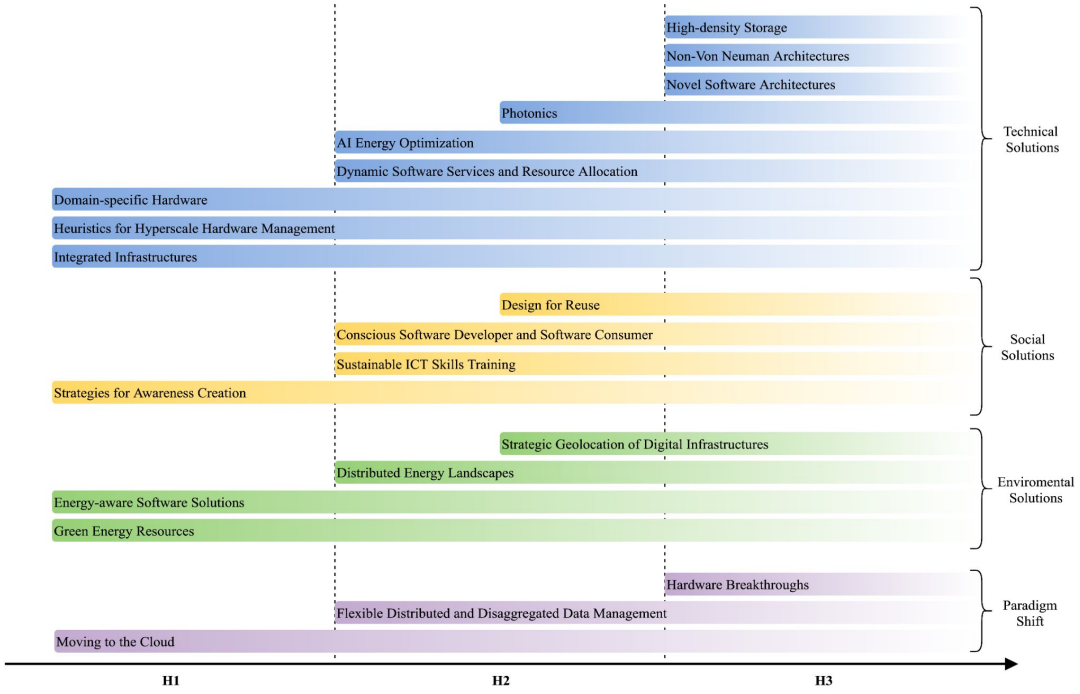

Figure 1: Solutions towards Green IT

Solutions for the Near Future:

With the near future we see our second paradigm shift (fig. 1, H2), namely flexible distributed and disaggregated data management. Supported by steady advancements in communication technologies and the growing affordability of computational power, a vast number of computational tasks will elastically move as near as possible to both the consumer premises and the increasingly decentralized energy producers, e.g., via onboard computing, and distributed networks of computational nodes. Solutions of H2 reflect the second paradigm shift and entail the appearance of distributed energy landscapes (e.g., prosumers), dynamic software services and hardware resource allocation, and the strategic positioning of digital infrastructures near to their end-users to ensure high band-width, keeping low communication energy consumption.

The ever-growing energy consumption of AI-based systems will be mitigated in the distributed paradigm of H2 with the rise of federated learning algorithms, affordable AI chips, and data usage plus compression strategies. Energy-aware software optimizations will continue to prosper, with consolidated tactics for software energy efficiency, seamless virtualization of hardware components, and energy-driven workload optimizations. From a social perspective, the growing demand for sustainable ICT skills will require training of new professional profiles. The awareness emerged in H1 will develop in H2 into conscious software developers and consumers, who will feel personally accountable for their actions, hence developing and demanding more sustainable solutions. Different from H1, where the average lifecycle expectancy of digital infrastructure hardware components equals approximately two years, H2 is also expected to witness a growing trend of design for hardware reuse and lifecycle management practices.

Further Evolution:

As we look further into the future, and uncertainty grows, we can foresee energy efficiency of digital infrastructures being supported by our last paradigm shift, namely novel hardware breakthroughs. While already supporting to a large extent the distributed and disaggregated paradigm, photonic technology will shape itself in H3 into integrated photonics, mitigating the culmination of micro-electronics computational advancements. In H3 non-von Neumann architectures (e.g., neuromorphic computing) will also become available for adoption, allowing to carry out complex computations at a fraction of their current energy cost. By making use of electron spins or relations between protons and electrons, high-density storage solutions will also become in use during H3. From a software perspective, to support the drastic hardware changes of H3, software systems will require novel software architectures, to best fit software artefacts to the new underlying hardware.

Green Software

IT practitioners could already start integrating them in their practice. Here we will provide some hands-on guidance for actually implementing green IT. Table 1 provides an overview with reference points.

Table 1: Examples of short-term solutions and related energy impacts

H1 Solution type	**Example of specific technique (if applicable)**	**Observed Energy Impact**	**Source**
Domain-specific Hardware	Utilizing AI accelerators	2.5% improvement of power performance every year	IBM. 2021. *Introducing the AI chip leading the world in precision scaling*}. URL: https://www.ibm.com/blogs/research/2021/02/ai-chip-precision-scaling.

Strategies for Awareness Creation	Providing actionable guidelines for data center operators to configure power management settings	10-13% energy savings	Netherlands Enterprise Agency, 2020. *Happy Flow manual 1.0. Energy-efficient design of data centers using power management and virtualization*. URL: https://amsterdameconomicboard.com/app/uploads/2020/10/Happy-Flow-Manual-LEAP.pdf.
Energy-Aware Software Optimizations	Shutting-down idle servers	84% of energy reduction	Asperitas. 2017. *The datacentre of the future*. URL: https://www.asperitas.com/whitepapers/the-datacentre-of-the-future.
Heuristics for Hyperscale Hardware Management	Replacing fan-based with liquid-based cooling	6-45% reduction of IT energy footprint	Rais *et al.* 2018. *Energy-saving potential of separated two-phase thermosiphon loops for data center cooling*. Journal of Thermal Analysis and Calorimetry.
Green Energy Resources	Replacing brown- with green-energy resources	100% consumed energy produced by renewable energy resources	Google. 2021. *Cloud Sustainability*. URL: https://cloud.google.com/sustainability.
Moving to the Cloud	Migrating from on-premises to the cloud	64% of energy reduction	Accenture. 2020. *Cloud Computing and Sustainability: The Environmental Benefits of Moving to the Cloud*. URL: https://www.accenture.com/us-en/insights/strategy/green-behind-cloud.

Software practitioners can contribute in many ways to accelerating the transition to an energy efficient digital infrastructure:

Move to the cloud. Cloud services typically decrease energy consumption compared to om-premise data centers due to less overheads and more efficiency in scalability. Depending on the sources, between 80% and 90% of the businesses have already migrated to the cloud [2,3]. This significant investment does not mean that “the job is done'' (also given that energy efficiency has hardly ever been a concern); rather, it revealed that effective and technically-sustainable migration demands major rearchitecting. This calls for the following concrete actions: When replacing on-premises functionality with cloud-native services, the services selection should be informed by transparent indicators (that must be provided by the cloud service providers) for the related energy- and resource efficiency; When re-engineering and re-architecting applications for the cloud, practitioners should include software tactics to manage data more efficiently to decrease the necessary resources and thus energy for data storage, computing resources to process data, and networking resources to transfer data between cloud-, edge- and customer sides. Examples of such energy efficient data management tactics may address, data flow optimization, data deduplication, smart data compression based on frequency of use. Serverless computing and functions-as-a-service (FaaS) facilitate seamless scalability of resource consumption. Data usage profiling and AI/ML techniques for profiling applications can discover data usage patterns and adjust performance towards more efficiency.

Ecology by design. Designers determine the ecologic footprint if their software. As shown in Figure 1, we will witness major paradigm shifts which demand an increased distribution of applications, data and computing resources between local-, edge- and remote cloud premises. To be ready for this shift, practitioners should already start to rearchitect their products with flexibility in mind. Embedding flexibility in software engineering,

like was never done before, allows to develop future proof software products, i.e., technically sustainable, as well as taking advantage of the promises of smart data staging and computational offload for delivering energy efficiency. As a designer ensure that new software versions do not require more computing power, hard-disk space, or bandwidth than before. This is easily feasible by optimizing algorithms. A simple checkpoint is that previous core functionality remains available on older hardware. As an embedded

Green user experience (UX). Users of software dominate the consumption because they scale eventually to the millions. Imagine the simple move from receiving high-resolution advertisement movies towards ad-blocking. This is the control of each user and will save both on bandwidth and energy consumption both in data centers, switches and not the least in each device, where less GPU power would be necessary. UX matters and can be made adjustable for less consumption. As a UX designer, make sure that users can configure software according to their individual needs, such as night mode, movie blocking, reduced resolution movies etc. UX should clearly show the power usage of the respective software. Create the awareness by showing with red/amber/green how much energy is currently consumed. This can be linked to processer usage etc. Make users able to manage the energy consumption of hardware as efficiently as possible, including energy-saving modes to easily switch down hardware when not used. This can be even automated by tracking usage and bringing unused processes to a sleep mode, or simply terminate those.

Blending software and its context. Creating energy-aware software and infrastructures calls for the much more explicit and pervasive inclusion of contextual information in design decision making. Context regards planning for known changes, adapting to unknown ones, and including systemic perspectives like different sectors, such as energy sector and ICT sector, stakeholders, such as the priorities, investments, and constraints of organizations, and systems, such as the role of data centers for hyper-scaling, and municipalities for micro-clouds and energy ecosystems in built environments. Let us look to blockchain technology as an example. Blockchains are increasingly used to establish electronic contracts, such as in finance transactions, replacement parts delivery in industry and aerospace, medical treatments, etc. Yet their energy hunger is growing in parallel. The major critique of any distributed ledger technology is its waste of energy to maintain the steady growing transaction history [7]. For instance, the Bitcoin currency gets more value primarily because it needs more computer power, which wastes energy. Blocks are validated using a proof-of-work riddle. Only the first peer to solve it gets rewarded, which has created a race for IT power to solve the riddles. Miners build clusters of specialized, optimized hardware. These clusters require power to operate and to cool them down. The energy needed to just maintain the Bitcoin currency is estimated to be beyond that of Switzerland [5]. As a software engineer try to avoid blockchain usage both in finance transactions and move towards alternative consensus techniques.

Competences. To move towards an energy efficient cloud, practitioners must embed strategies for awareness creation for measuring, monitoring, and visualizing the energy footprint of the software they deliver. The software energy consumption must be measured, estimated, and predicted at all levels, from the used computing resources like CPU/GPU, to the generated data traffic; from the single cloud-native service and container, to whole systems and apps. Education and training programs must equip new talents and the existing workforce with the related know-how.

Ecologic hardware. Continuously demanding, installing, and using the latest ad highest-performing hardware highly contributes to both e-waste and energy consumption. When designing software, consider that it can also be deployed previous hardware platforms. Aging is normal but should not be measured in months but rather several years. Embedded designers are aware of the problem and have the solutions. When they design a piece of software, for avionics, industry automation or simple upgradeable IoT devices, they decouple the software from the underlying hardware, thus allowing the hardware being used for many years. Over-the-air upgrades can introduce new and fancy features, but also security patches and error corrections – without demanding new hardware. For instance, mobile vehicles from automotive up to trains and airplanes are typically running for years if not decades on the same platform by simply upgrading some controllers towards new hardware, while keeping the entire system active. A car typically has a life-time of ten years where it will occasionally get new hardware devices, if the previous set has reached end of life. Trains and other industry systems have much longer life-time. So why would we need buying new consumer goods on a yearly base? To reduce your own personal ecologic

footprint consider buying refurbished equipment. It is a simple yet effective step for your own ecologic contribution.

Ecologic Behaviors Are the New Normal

Ecologic transformation is the next big wave of innovation after the digital transformation. Every business will evolve towards ecologic behaviors. Investors are currently changing behaviors towards financing based on ecology and sustainability. They not only do that for ethical reasons, but simply because economic risks of traditional business are too high. Another driver is market changes in behaviors across countries, worldwide. This currently still looks unbalanced with some parts of Europe being rather advanced, while other countries stay in old-fashioned energy-wasting behaviors. Yet, it will change, if people not only look towards low pricing, but also ecology and sustainability in a new normal. Buying power will have increasingly impact on any business.

Behaviors matter for ecologic transformation. For one part it means to evangelize in our respective spheres. During the past two years, online collaboration has exploded to mitigate pandemic risks. Tools such as Skype, Teams and Zoom have replaced the traditional meet and greet – at no cost and with big wins on efficiency and flexibility. Traveling suddenly has become a legacy with its high travel time and cost overheads, combined with huge ecologic footprint. While meeting real people is still good to build relationships, online meetings allow to adjust instantaneously to specific needs. In the recent survey of Vector and IEEE Software a vast majority of over 90% of respondents acknowledge the advantages of online meetings and trainings [4].

Digital transformation has moved behaviors from traditional business to IT-driven business. Yet is has the drawback of currently wasting huge amounts of energy. Green IT and green coding are the counter-mechanisms which address the root cause by reducing energy consumption of our IT and software systems. A good showcase of how fast this can change are our laptops. While the energy consumption had been stable for a single laptop over the past three decades, their power and performance had improved by the millions. This was possible by advanced hardware design and dynamically switching off those arts which are currently idle or unnecessary. The same holds for the on-board software of vehicles which stabilized around few kilowatts in some fifty plus controllers, and currently is decreasing. The immediate benefits are obvious due to extended battery life and less cooling needs. Obviously, we will move to green IT for more reasons than only ecology, but ecology will accelerate the transformation.

Too often and wrongly ecology and economy are perceived as antagonists. Ecologic behaviors and climate protection are not a threat, but a great business opportunity. Today, environmental protection is driven by technology, innovation, and smart IT. It is not slogans alone, but us ordinary engineers who facilitate te ecology transformation by deploying green IT. Famous American politician and scientist Benjamin Franklin once remarked: “If you fail to prepare, you prepare to fail.”. Green IT and ecologic business is today’s answer towards saving the planet.

Sidebar: Green Testing

Testing consumes not only most of the time and effort in a software project, but also heavily contributes to energy waste. Testers know that it is not the amount of testing but the quality of testing which matters. Still, we see in our consulting practice many test teams that pile up test cases without any underlying test strategy. With a client we have reworked a test strategy which was built upon a huge amount of brute force testing towards intelligent testing.

Previous testing followed the traditional scheme of collecting test cases. The low-level code testing used a Jenkins pipeline with automatic unit test cases. Above was an integration test which ensured that functions and interfaces worked properly. The system test was a more experience-driven set of test clusters for base functions, UX and networking to other components of a yet bigger system. So far so unstructured. All three test layers were

basically collections of test cases, which were as much as possible automated, but not controlled with test methods such as coverage, risk-orientation, heat maps, change logs and traceability for regression test. Test share compared to total project effort was growing by the year from initially thirty percent towards over fifty percent when we started. Test effort in such brute force scheme is highly correlated with test energy consumption which was visible not the least in the amount of cooling necessary in summertime for their on-premises test equipment.

As a first step we introduced test methodologies and systematically structured the test process. For instance, we introduced static code analysis to identify hot spots which need more low-level testing. Test cases and test management was instrumented to identify test effectiveness and test efficiency. This allowed to substantially reduce test cases by removing redundancies. At the same time test effectiveness was improved by targeting previously uncovered functions and code segments. Test-driven methods were introduced to start from the beginning of a change or new functionality with testability in mind. Test-Driven Development (TDD) facilitates not only a basic set of unit test cases, but also ensures better code quality. Test-Driven Requirements Engineering (TDRE) specifies functions as test cases. Both TDD and TDE facilitate a much more efficient regression testing, which with many changes over the project-time and product life-time saves lots of effort – end energy.

In a next step we introduced risk-oriented testing. In a distributed system, undesirable behavior and malfunctions can arise because there has been a software change at one point that breaks through to other components. This raises numerous questions: How can the function of a system be ensured if changes take place in the subcomponents? How can the safety and reliable behavior be guaranteed if software changes are made to individual components during operation? Unlike brute force with low efficiency, risk-oriented testing evaluates the dependencies of functions and code, both internally, i.e., white-box, as well as externally, i.e., black-box. An external dependency is for instance a critical functionality such as billing of a finance service or a safety-critical function in a robot.

Finally, we introduced intelligent testing with AI to support validation. There is many AI-based test methods, ranging from rule-based systems, fuzzy logic, Bayesian nets to the multiple neural network approaches of deep learning. The potential for an intelligent testing is manifold: On a system level there are questions on which test cases must be executed, and to what extent? Intelligent validation, such as cognitive testing, helps in selecting and creating test cases. In a first step an assistance functionality is set up to identify priorities in an existing set of use cases. As a result, the validation expert can test quicker and with a better coverage of situational relevant scenarios. On the level of a component or module testing it is also required to identify relevant cases. This can range from a simple support on how to feed the system with scenario-based data generation and check on the outputs with a test-oracle, towards complex algorithms which automatically create test cases based on the code or user interface.

As a result, test effort was reduced while at the same time improving test effectiveness. Given the high amount of test equipment and its steady energy consumption, we achieved a double-digit reduction in electrical energy.

Side-Bar 2: Industry Survey

This column will furtheron address major challenges in Software and IT. With the current fast changing software and IT landscape, you are invited to take part in the short survey on industry trends 2022. Your opinion as a domain expert and decision maker is valuable. Please give us two minutes of your time and answer four brief questions. After the end of the survey, you will exclusively receive its analysis, enriched with experiences and hands-on advice from current projects. With some luck you will also receive a free copy of the IEEE book “Global Software and IT”. We will not use your personal data any further. Here the link to the industry survey:
www.vector.com/trends-survey

References

[1] Verdecchia, R., Lago, P., de Vries, C., 2021. “LEAP Technology Landscape: trends and scenarios”. Persistent URL: https://tinyurl.com/leapLandscape . Accessed: 1. Aug. 2021.
[2] Desjardins, J., Ang, C., Lu, M., 2014. Cloud Computing Growth - Visual Capitalist. https://www.visualcapitalist.com/cloud-computing-growth/ . Accessed: 1. Aug. 2021.
[3] Galov, N., 2020. 25 Cloud Computing Statistics in 2020 - Will AWS Domination Continue? https://hostingtribunal.com/blog/cloud-computing-statistics/ . Accessed: 1. Aug. 2021.
[4] Ebert, C. and B.Tavernier: “Technology Trends: Strategies for the New Normal”. IEEE Software, vol. 38, no. 2, pp 7–14, ISSN: 0740-7459, Mrc. 2021.
[5] Podder, S., A. Burden, S. Kumar Singh, and R. Maruca: “How Green Is Your Software?”. Harvard Business Review, September 2020, https://hbr.org/2020/09/how-green-is-your-software . Accessed: 1. Aug. 2021
[6] Ebert, C. and R.Ruschil: „Test-Driven Requirements Engineering”. IEEE Software, ISSN: 0740-7459, vol. 38, no. 1, pp. 16-24, Jan. 2021.
[7] Ebert, C., P. Louridas, T.M. Fernández-Caramés, and P.Fraga-Lamas: „Blockchain Technologies in Practice“. IEEE Software, ISSN: 0740-7459, vol. 37, no. 3, pp. 17-25, July 2020

Acknowledgments

Some parts of this research received funding from the Netherlands Enterprise Agency Project ``Energy Efficient Digital Infrastructures'' (project number RVO-TSE2200010) and support from the LEAP Initiative of the Amsterdam Economic Board.

Authors

Roberto Verdecchia is a Research Associate at the Software and Sustainability group of the Vrije Universiteit Amsterdam, 1081 HV Amsterdam, the Netherlands. More information is available at https://robertoverdecchia.github.io. Contact him at r.verdecchia@vu.nl

Patricia Lago is full professor in software engineering at the Vrije Universiteit Amsterdam, 1081 HV Amsterdam, the Netherlands, where she leads the Software and Sustainability research group in the Computer Science Department. She is a co-founder of the Green Lab, More information is available at www.patricialago.nl Contact her at pdotlago@gmail.com

Christof Ebert is the managing director of Vector Consulting Services, Stuttgart, 70499, Germany. He serves on the editorial board of IEEE Software and is a Senior Member of IEEE. Further information about him can be found at https://twitter.com/christofebert. Contact him at christof.ebert@vector.com.

Carol de Vries is program and technology manager at Photondelta in 5656 AA Eindhoven, the Netherlands. His focus is on integrated photonics technologies and applications and creating more business opportunities in this emerging field. Contact him at Carol@photondelta.eu